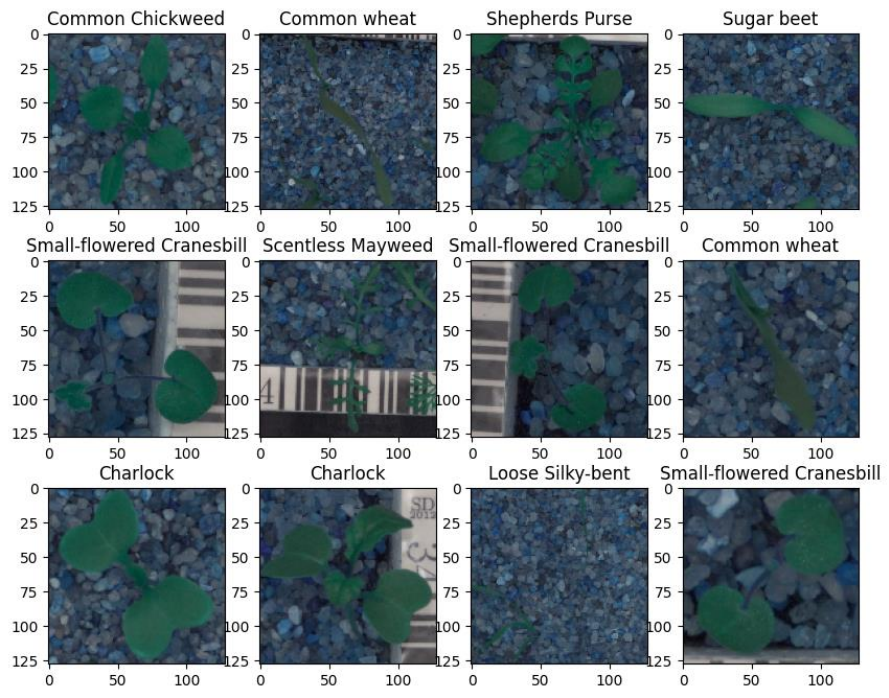
PLANTS SEEDLING CLASSIFICATION

Tyler Walje



OBJECTIVE

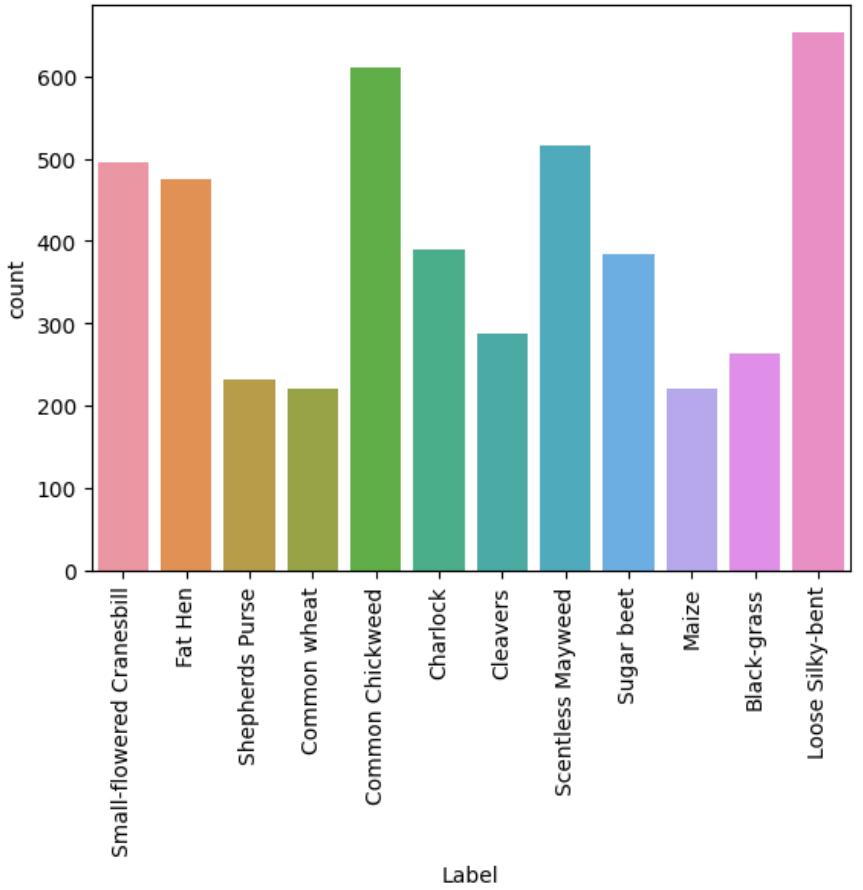
In order to reduce the manual labor and workload for workers in the agriculture industry, we can develop a CNN (Convolutional Neural Network) to identify plant seedlings. This, hopefully, will greatly reduce manual labor, increase the accuracy identifying specific plants, and free up time and energy for the workers in this industry.

In order to build the CNN, we will use a dataset that includes many photos of the 12 different species of plants. We'll use these pictures to train the model which will be ultimately used to identify the plant seedlings.

Possible Setbacks

During the data exploration, we see that some pictures of plants are more abundant than other pictures. For example, there is about 3x the amount of “Loose Silky-bend” and “Common Chickweed” compared to “Maize”, “Common Wheat” and “Shepherds Purse”.

The background of these pictures might lead to problems, so we’ll have to adjust the images from BGR to RGB to help mitigate for this.



PREPARING TO BUILD THE MODELS

1. Convert images from BGR to RGB
2. Resize images
3. Check the shape of the training, validation and test data
4. Encode target labels

First Model

- The first model will include 128 filters and a kernel size of 3x3
- After pooling we'll flatten the images
- Add a fully connected layer with 16 neurons and an output layer with, obviously, 12 neurons
- Next, we fit the model to the training data and run 30 epochs

SUMMARY OF FIRST MODEL

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 64, 64, 128)       3584

 max_pooling2d (MaxPooling2D  (None, 32, 32, 128)      0
 )

 conv2d_1 (Conv2D)           (None, 32, 32, 64)        73792

 max_pooling2d_1 (MaxPooling  (None, 16, 16, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 16, 16, 32)        18464

 max_pooling2d_2 (MaxPooling  (None, 8, 8, 32)         0
 2D)

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 16)                32784

 dropout (Dropout)           (None, 16)                0

 dense_1 (Dense)             (None, 12)                204

=================================================================
Total params: 128,828
Trainable params: 128,828
Non-trainable params: 0
```

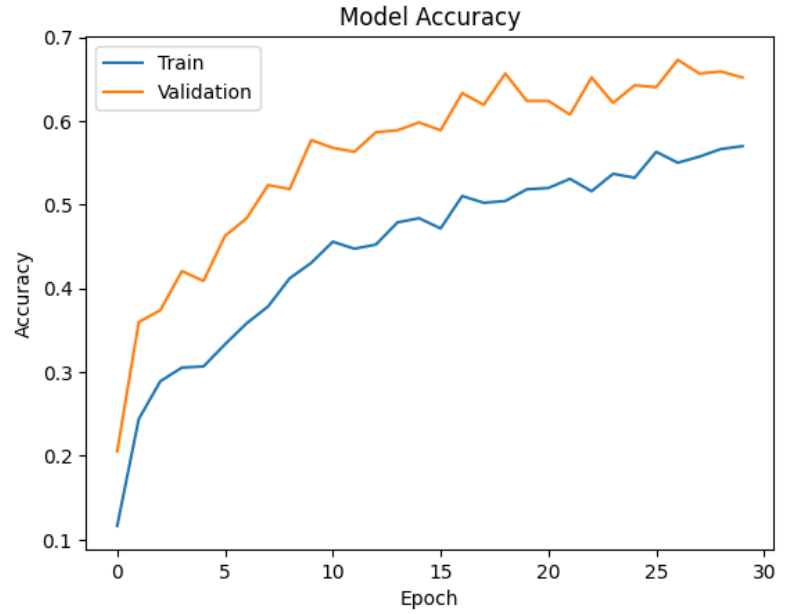
First Model

Evaluating the Model

The 1st model appears to be underfit. We see this in the graph to the right. The validation test set somewhat mirrors the training set, as well as the training loss continuously decreasing until the end of training.

We can see that the accuracy is improving though. We seem to be on the right path.

ACCURACY OF FIRST MODEL



LAST 10 EPOCHS
(showing decreasing training loss)

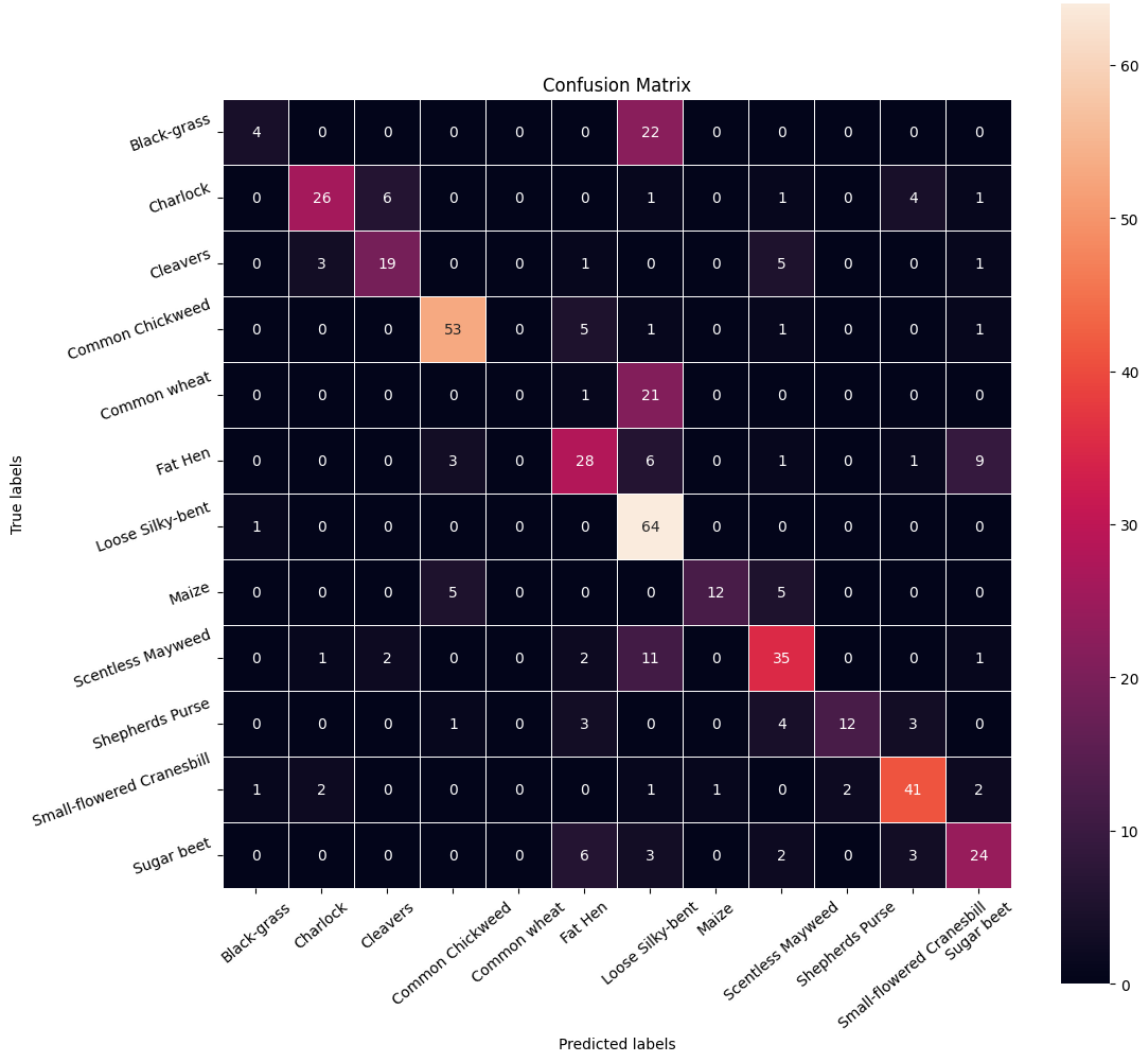
```
Epoch 21/30
121/121 - 1s - loss: 1.2692 - accuracy: 0.5199 - val_loss: 1.1460 - val_accuracy: 0.6238 - 1s/epoch - 11ms/step
Epoch 22/30
121/121 - 1s - loss: 1.2470 - accuracy: 0.5308 - val_loss: 1.2255 - val_accuracy: 0.6075 - 1s/epoch - 10ms/step
Epoch 23/30
121/121 - 1s - loss: 1.2647 - accuracy: 0.5160 - val_loss: 1.1041 - val_accuracy: 0.6519 - 1s/epoch - 10ms/step
Epoch 24/30
121/121 - 1s - loss: 1.2249 - accuracy: 0.5368 - val_loss: 1.1764 - val_accuracy: 0.6215 - 1s/epoch - 9ms/step
Epoch 25/30
121/121 - 1s - loss: 1.2139 - accuracy: 0.5321 - val_loss: 1.0813 - val_accuracy: 0.6425 - 1s/epoch - 9ms/step
Epoch 26/30
121/121 - 1s - loss: 1.1824 - accuracy: 0.5630 - val_loss: 1.0996 - val_accuracy: 0.6402 - 1s/epoch - 9ms/step
Epoch 27/30
121/121 - 1s - loss: 1.1735 - accuracy: 0.5500 - val_loss: 1.0667 - val_accuracy: 0.6729 - 1s/epoch - 9ms/step
Epoch 28/30
121/121 - 1s - loss: 1.1559 - accuracy: 0.5573 - val_loss: 1.0755 - val_accuracy: 0.6565 - 1s/epoch - 9ms/step
Epoch 29/30
121/121 - 1s - loss: 1.1582 - accuracy: 0.5664 - val_loss: 1.0585 - val_accuracy: 0.6589 - 1s/epoch - 9ms/step
Epoch 30/30
121/121 - 1s - loss: 1.1301 - accuracy: 0.5698 - val_loss: 1.1177 - val_accuracy: 0.6519 - 1s/epoch - 10ms/step
```

First Model

Evaluating the Model

We need more data (pictures). Some plants were commonly mis-identified or not identified. “Common Wheat” didn't correctly identify even one picture and was commonly mis-identified as “Loose Silky-bent”. Speaking of “Loose Silky-bent”, while it was the most abundant picture that was recognized, it also mis-classified as “Scentless Mayweed”, “Common Wheat” and “Black-grass”. The model seems to be effective identifying several plants including “Charlock”, “Cleavers”, “Common Chickweed” and “Small-flowered Cranesbill”.

CONFUSION MATRIX OF FIRST MODEL



First Model

Evaluating the Model

Graphic on this slide: index of all 12 plant species

Precision: The weighted average precision was 67%. Some didn't perform well (4 was never correctly predicted) and some performed great (7 was correctly predicted 92% of the time)

Recall: Wide range of recall with the lowest again being class 4 (0%) and the highest being class 6 (98%). Average is 67%.

Support: We need more data and more pictures. We also need more balance in the dataset.

CLASSIFICATION REPORT OF FIRST MODEL

	precision	recall	f1-score	support
0	0.67	0.15	0.25	26
1	0.81	0.67	0.73	39
2	0.70	0.66	0.68	29
3	0.85	0.87	0.86	61
4	0.00	0.00	0.00	22
5	0.61	0.58	0.60	48
6	0.49	0.98	0.66	65
7	0.92	0.55	0.69	22
8	0.65	0.67	0.66	52
9	0.86	0.52	0.65	23
10	0.79	0.82	0.80	50
11	0.62	0.63	0.62	38
accuracy			0.67	475
macro avg	0.66	0.59	0.60	475
weighted avg	0.67	0.67	0.65	475

MODEL 1 TAKEAWAYS

1. The Model seems to be underfit
    1. The training loss decreased consistently through training
    2. The shape of the train/validation suggests an underfit model
2. We need more data/pictures of the plants
    1. There was an imbalance of pictured plants to begin with
    2. With more data, we might get closer to more accurate results
3. The model did a decent job of identifying *some* plants
    1. There were some plants that did very well in some areas (class 6 had a 98% recall) and other plants that did very poorly (class 4 had a 0% recall)

Second Model

- In the 2nd model, we'll reduce the learning rate with ReduceLROnPlateau().
- The second model will include 64 filters (compared to 128 on the first model) and a kernel size of 3x3
- After pooling we'll flatten the images
- Add a fully connected layer with 16 neurons and an output layer with, obviously, 12 neurons
- Next, we fit the model to the training data and run 30 epochs

SUMMARY OF SECOND MODEL

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 64, 64, 64)        1792

 max_pooling2d (MaxPooling2D  (None, 32, 32, 64)       0
 )

 conv2d_1 (Conv2D)           (None, 32, 32, 32)        18464

 max_pooling2d_1 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 batch_normalization (BatchN  (None, 16, 16, 32)       128
 ormalization)

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 16)                131088

 dropout (Dropout)           (None, 16)                0

 dense_1 (Dense)             (None, 12)                204

=================================================================
Total params: 151,676
Trainable params: 151,612
Non-trainable params: 64
_________________________________________________________________
```

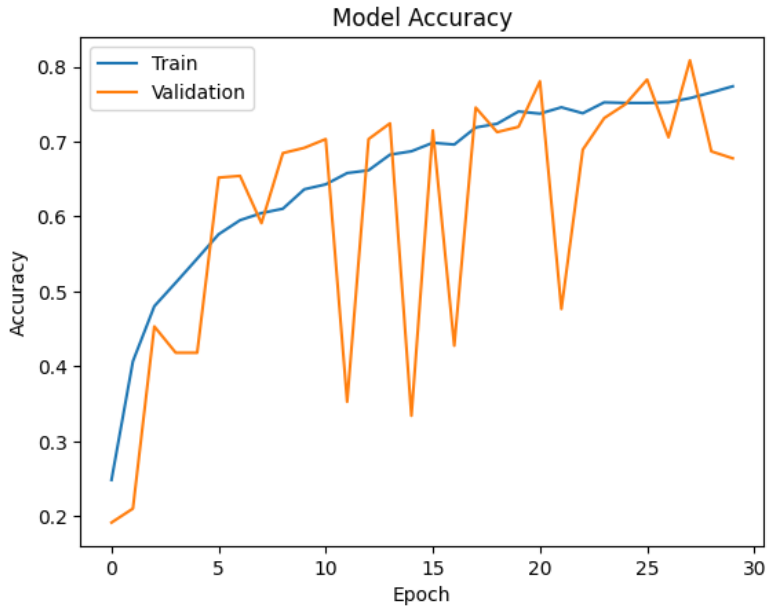
Second Model

Evaluating the Model

The 2nd model appears to be unrepresentative. We see this in the graph to the right. The validation set is all over the place and shows signs of spiking up and down. This indicates noise and usually means that the dataset does not provide enough useable information.

While the accuracy is improving, this graph does not look great. It is potentially due to the reduced learning rate or the reduced number of filters.

ACCURACY OF SECOND MODEL



LAST 10 EPOCHS
(showing reduced learning rate at epoch 24)

```
Epoch 21/30
60/60 [==============================] - 4s 74ms/step - loss: 0.7002 - accuracy: 0.7372 - val_loss: 0.7348 - val_accuracy: 0.7804 - lr: 5.0000e-04
Epoch 22/30
60/60 [==============================] - 4s 74ms/step - loss: 0.6939 - accuracy: 0.7457 - val_loss: 1.8701 - val_accuracy: 0.4766 - lr: 5.0000e-04
Epoch 23/30
60/60 [==============================] - 6s 97ms/step - loss: 0.6972 - accuracy: 0.7378 - val_loss: 1.1427 - val_accuracy: 0.6893 - lr: 5.0000e-04
Epoch 24/30
60/60 [==============================] - ETA: 0s - loss: 0.6896 - accuracy: 0.7523
Epoch 24: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
60/60 [==============================] - 4s 74ms/step - loss: 0.6896 - accuracy: 0.7523 - val_loss: 0.8683 - val_accuracy: 0.7313 - lr: 5.0000e-04
Epoch 25/30
60/60 [==============================] - 6s 100ms/step - loss: 0.6787 - accuracy: 0.7515 - val_loss: 0.8291 - val_accuracy: 0.7500 - lr: 2.5000e-04
Epoch 26/30
60/60 [==============================] - 5s 75ms/step - loss: 0.6567 - accuracy: 0.7515 - val_loss: 0.7288 - val_accuracy: 0.7827 - lr: 2.5000e-04
Epoch 27/30
60/60 [==============================] - 6s 103ms/step - loss: 0.6614 - accuracy: 0.7523 - val_loss: 0.9496 - val_accuracy: 0.7056 - lr: 2.5000e-04
Epoch 28/30
60/60 [==============================] - 4s 74ms/step - loss: 0.6442 - accuracy: 0.7579 - val_loss: 0.7301 - val_accuracy: 0.8084 - lr: 2.5000e-04
Epoch 29/30
60/60 [==============================] - 4s 74ms/step - loss: 0.6329 - accuracy: 0.7655 - val_loss: 0.9892 - val_accuracy: 0.6869 - lr: 2.5000e-04
Epoch 30/30
60/60 [==============================] - 6s 101ms/step - loss: 0.6245 - accuracy: 0.7737 - val_loss: 1.1309 - val_accuracy: 0.6776 - lr: 2.5000e-04
```

Second Model

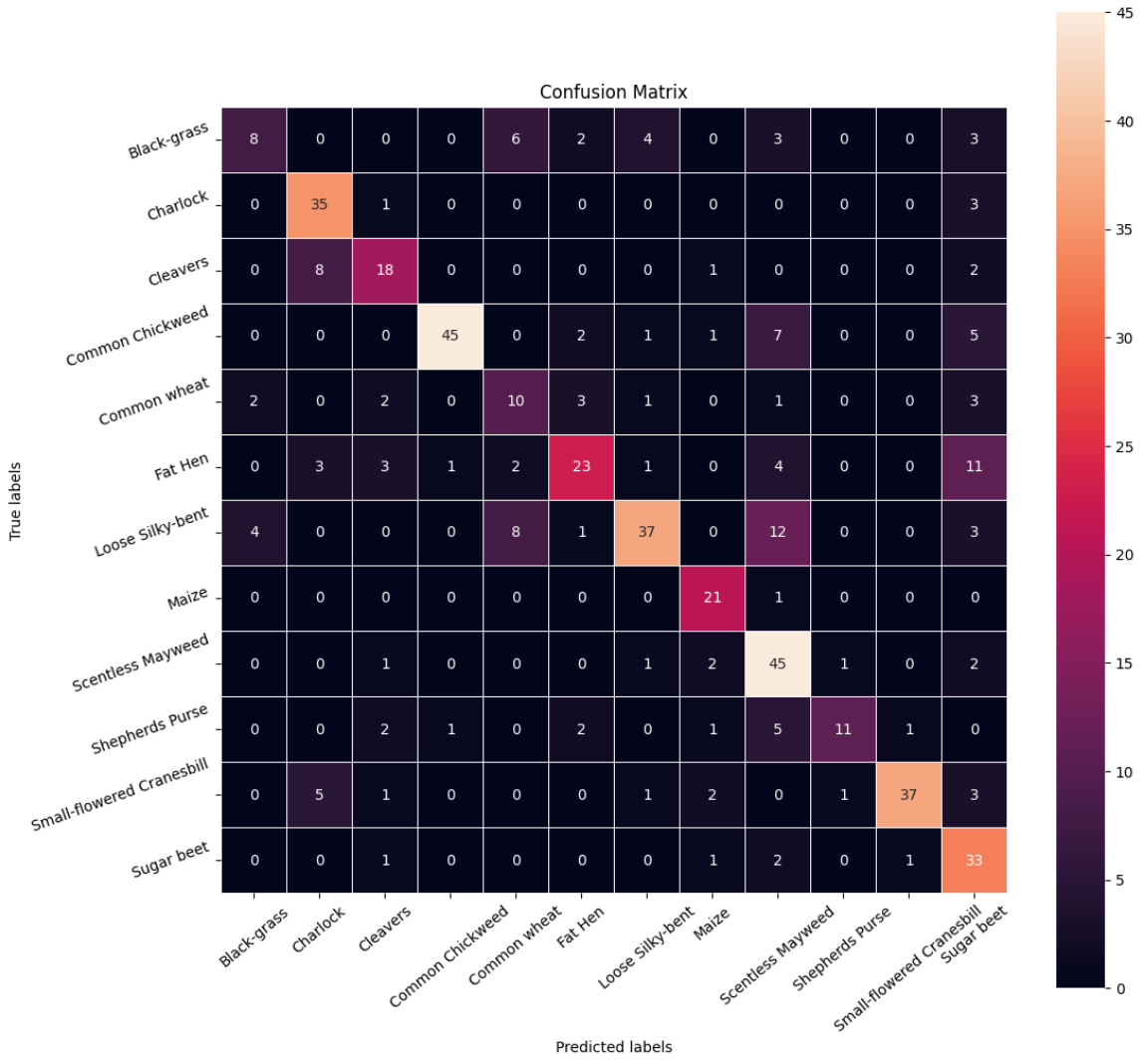
Evaluating the Model

This confusion matrix shows some improvement. The train/validation graph was not promising, but this seems to be better.

We see improvement with the mis-identification of “Loose Silky-bent” which seemed to be an issue in the first model. We are also seeing an increase in accuracy with “Black-grass” as well as “Common wheat”.

“Small-flowered Cranesbill Sugar beet” incorrectly predicted “Fat Hen” 11 times, which was slightly worse than Model 1.

CONFUSION MATRIX OF SECOND MODEL



Second Model

Evaluating the Model

Graphic on this slide: index of all 12 plant species

Precision: The weighted average precision was 72%. We saw an overall increase in the precision, but it was minimal.

Recall: The weighted average was only 1% better (from 67% to 68%) so not much improvement. Still a wide range with the lowest being 31% and the highest being 95%. Overall, there seems to be improvement with the variance of the results.

Support: We need more data and more pictures. We also need more balance in the dataset.

CLASSIFICATION REPORT OF SECOND MODEL

	precision	recall	f1-score	support
0	0.57	0.31	0.40	26
1	0.69	0.90	0.78	39
2	0.62	0.62	0.62	29
3	0.96	0.74	0.83	61
4	0.38	0.45	0.42	22
5	0.70	0.48	0.57	48
6	0.80	0.57	0.67	65
7	0.72	0.95	0.82	22
8	0.56	0.87	0.68	52
9	0.85	0.48	0.61	23
10	0.95	0.74	0.83	50
11	0.49	0.87	0.62	38
accuracy			0.68	475
macro avg	0.69	0.66	0.65	475
weighted avg	0.72	0.68	0.68	475

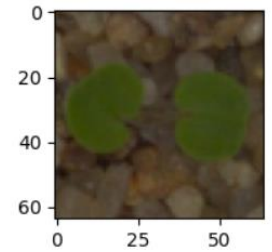
MODEL 2 TAKEAWAYS

1. The Model seems to be unrepresentative
    1. The training loss graph showed noisy movements in the validation training set
    2. The shape of the train/validation suggests an unrepresentative model
2. We still need more data/pictures of the plants
    1. There was an imbalance of pictured plants to begin with
    2. With more data, we might get closer to more accurate results
3. The model did a decent job of identifying *some* plants and there were overall improvements from the outliers in Model 1
    1. In Model 1, class 4 had a 0% recall. That improved to 45% in Model 2.
4. There were overall some issues in the 2nd Model and I believe more needs to be done, however, there were enough improvements to believe we are making progress with the 2nd Model.

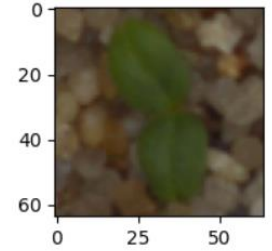
Final Model

- The final model correctly identified 4 different plants out of 4 opportunities (shown on the right).
- We are certainly getting closer to a model that can replace agricultural workers.
- Despite the concerning numbers before, it is reassuring to see all 4 plants identified correctly.

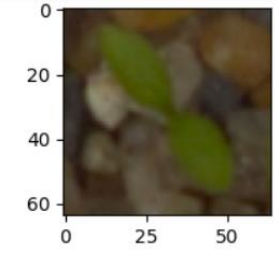
FINAL MODEL PREDICTIONS AND RESULTS



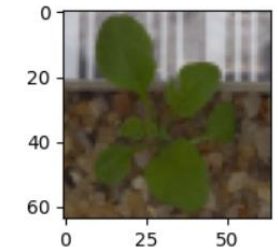

```
1/1 [==============================] - 0s 163ms/step
Predicted Label ['Small-flowered Cranesbill']
True Label Small-flowered Cranesbill
```




```
1/1 [==============================] - 0s 32ms/step
Predicted Label ['Cleavers']
True Label Cleavers
```




```
1/1 [==============================] - 0s 43ms/step
Predicted Label ['Common Chickweed']
True Label Common Chickweed
```




```
1/1 [==============================] - 0s 90ms/step
Predicted Label ['Shepherds Purse']
True Label Shepherds Purse
```

BUSINESS RECOMMENDATIONS

- In order for Artificial Intelligence to assist workers in the agricultural industry, more work needs to be done.
- We are off to a great start building a model to identify plant seedlings, but are still far off from something that could effectively and efficiently replace a human.
- More pictures are needed in the dataset. My recommendation would be to provide lots more data (pictures of plants). This would help to build better models and give us more leeway to manipulate the data and test/train the data.

BUSINESS RECOMMENDATIONS

- While the Final Model showed the correct plants 4 out of 4 times, I believe that this model is not yet operating as efficiently as it could be. It is therefore recommended that agricultural industry workers continue their current work schedule and workload as we continue to improve the model.
- When a model is built that CAN replace a human, it is highly recommended that it is immediately implemented. This will give workers more time to spend more effectively. This will ALSO provide the benefit of identifying plants correctly more often and more efficiently.

THANK YOU

Tyler Walje